

Тестирование API: <https://reqres.in/>

Из ходя из возможностей api <https://reqres.in/api/> можно разбить тесты на следующие модули:

- conn — проверка соединения и возможности тестов корректно обработать exception.ReadTimeout с помощью параметра ?delayed=timeout
- get — тестирование запросов для GET метода
- post — тестирование запросов для POST метода
- put — тестирование запросов для PUT метода
- patch — тестирование запросов для PATCH метода
- delete — тестирование запросов для DELETE метода

Из ходя из этого, можно выделить три уровня приоритета тестов(в терминологии allure.severity_level):

CRITICAL — conn (наиболее приоритетные кейсы для проверки работоспособности сервиса)

NORMAL — post, put, patch, delete (кейсы проверки корректности записи, обновления, удаления данных, с учётом того, что это, по сути, эхо сервер)

MINOR — get (наименее приоритетные кейсы для проверки корректности и возможности отдавать данные клиенту)

Далее модули подробнее:

conn

Тест-кейсы:

1. Проверка соединения — должен ответить кодом 200 на запрос вида reqres.in/api/anywords

2. Проверка таймаута

Запрос	timeout	Статус
?delay=-1	5	200
?delay=0	5	200
?delay=3	5	200
?delay=10	5	exception

post

Тест-кейсы:

3. Создание

Запрос	Данные	Ожидаемый статус
/api/users	{'name': 'morpheus','job': 'leader'}	201
/api/users	{'name': '', 'job': ''}	201
/api/users	None	201

4. Логин

Запрос	Данные	Ожидаемый статус
/api/login	{'email': 'peter@klaven', 'password': 'cityslicka'}	200
/api/login	{'email': 1, 'password': 2}	200
/api/login	{'email': 'peter@klaven'}	400
/api/login	{'email': None, 'password': None}	400
/api/login	{'password': 'pistol'}	400

5. Регистрация

Запрос	Данные	Ожидаемый статус
/api/register	{'email': 'peter@klaven', 'password': 'cityslicka'}	201
/api/register	{'email': 1, 'password': 2}	201
/api/register	{'email': 'peter@klaven'}	400
/api/register	{'email': None, 'password': None}	400
/api/register	{'password': 'pistol'}	400

put

Тест-кейсы:

По логике метода PUT тут должна быть проверка на соответствие данных URI, но поскольку это эхо сервер, то этого не происходит.

6. Обновление:

Запрос	Данные	Ожидаемый статус
/api/users	{'name': 'morpheus', 'job': 'leader'}	200
/api/users/-1	{'name': 'morpheus', 'job': 'leader'}	200
/api/users/0	{'name': 'morpheus', 'job': 'leader'}	200
/api/users/10	{'name': 'morpheus', 'job': 'leader'}	200
/api/users/100	{'name': 'morpheus', 'job': 'leader'}	200
/api/users/100	None	200

patch

Тест-кейсы:

7. Обновление

Запрос	Данные	Ожидаемый статус
/api/users	{'name': 'morpheus','job': 'leader'}	200
/api/users/-1	{'name': 'morpheus','job': 'leader'}	200
/api/users/0	{'name': 'morpheus','job': 'leader'}	200
/api/users/10	{'name': 'morpheus','job': 'leader'}	200
/api/users/100	{'name': 'morpheus','job': 'leader'}	200
/api/users/100	None	200

delete

Тест-кейсы:

8. Удаление

Запрос	Данные	Ожидаемый статус
/api/users	{'name': 'morpheus','job': 'leader'}	204
/api/users/-1	{'name': 'morpheus','job': 'leader'}	204
/api/users/0	{'name': 'morpheus','job': 'leader'}	204
/api/users/10	{'name': 'morpheus','job': 'leader'}	204
/api/users/100	{'name': 'morpheus','job': 'leader'}	204
/api/users/100	None	204

get

Тест-кейсы:

9.Запрос данных одного пользователя/некоторого ресурса

Запрос	Данные	Ожидаемый статус
/api/users/-1(/api/resource/-1)		404
/api/users/0(/api/resource/0)		404
/api/users/100(/api/resource/100)		404
/api/users/foo(/api/resource/foo)		404
/api/users/bar(/api/resource/bar)		404
/api/users/1(/api/resource/1)		200
/api/users/12(/api/resource/12)		200

10.Запрос данных о нескольких пользователях/некоторых ресурсах

Запрос	Данные	Ожидаемый статус
/api/users?page=-1(/api/resource?page=-1)		200
/api/users?page=0(/api/resource?page=0)		200
/api/users?page=2(/api/resource?page=2)		200
/api/users?page=4(/api/resource?page=4)		200
/api/users?page=foo(/api/resource?page=foo)		200
/api/users?page=bar(/api/resource?page=bar)		200
/api/users?page=10(/api/resource?page=10)		200(но при этом данные отсутствуют)
/api/users?page=10(/api/resource?page=10000)		200(но при этом данные отсутствуют)

Итого: получилось 3 уровня приоритетов, 6 модулей, 10 тест-кейсов, 64 теста. Во многих тестах из-за нелогичного поведения сервиса не было возможности взять граничные значения, поэтому количество тестов несколько избыточно.